

ЗАДАНИЕ 1 (РАБОТА С ФАЙЛОВЫМ РЕСУРСОМ)

1. Создать в файл `form.php`, который будет содержать в себе форму с одним текстовым полем (с именем поля, например, `content`), а также логику обработки этой формы.
2. Открыть командную строку, перейти в ней в директорию, где находится файл и запустить команду `php -S localhost:8000`
3. После заполнения формы (<http://localhost:8000/form.php>) и отправки данных на сервер, PHP код должен записать текстовое поле (из формы) в файл `info.txt` при помощи функций `fopen` и `fwrite`. Если файл уже существует, то он должен не перезаписываться, а дополняться, т.е. старые сохранённые текстовые поля остаются, а каждое новое поле записывается с новой строки (символ новой строки - `\n` - чтобы он нормально воспринимался, помещайте его в двойные кавычки).
4. Не забываем закрывать ресурс (`fclose`) после работы с ним.
5. Дополнительное условие: если файл `info.txt` уже существует - каждый раз при заходе на страницу открыть ресурс (`fopen`), ссылающийся на него, а затем выбирать построчно все данные и отображать в виде html таблицы (1 строка - 1 ряд таблицы). Проверку на существование файла `info.txt` (если обратиться к нему для чтения, а его не окажется - будет ошибка) надо осуществлять при помощи функции `file_exists`:

```
<?php
$lines = [];
if (file_exists('info.txt')) {
    // какая-то логика
}
```

Пример таблицы:

```
<table>
  <tr>
    <td>1-я колонка 1-го ряда</td>
  </tr>
  <tr>
    <td>1-я колонка 2-го ряда</td>
  </tr>
</table>
```

Пример формы:

```
<form action="form.php" method="POST">
  Content:
  <br>
  <textarea name="content"></textarea>
  <br><br>
  <button type="submit" name="submit-form" value="1">
    Submit
  </button>
</form>
```

ЗАДАНИЕ 2 (ОБХОД ДИРЕКТОРИИ)

1. Создайте файл `dir.php`, который будет отображать структуру какой-либо директории (всех файлов, вложенных директорий, файлов вложенных директорий и т.д.).
2. Открыть командную строку, перейти в ней в директорию, где находится файл и запустить команду `php -S localhost:8000`
2. Для отображения (<http://localhost:8000/dir.php>) создайте функцию `parseDir`, которая принимает строку – путь к необходимой директории (подставьте какую-либо директорию – не очень глубокую, которая находится на вашем компьютере).
3. Структура директории должна отображаться в виде html списка (`ul`), кроме того – директории должны быть обрاملены в тег ``, а файлы – в тег `<i>`
4. Для получения содержимого директории используйте функцию `scandir`, для проверки на файл – `is_file`, а для проверки на директорию – `is_dir`
5. Помните о том, что `.` – запрашиваемая директория, а `..` – родительская директория – если их не игнорировать, то ваш код будет выполняться бесконечно!
6. Не стесняйтесь вызывать функцию `parseDir` внутри самой функции `parseDir` – ведь внутри каждой директории могут быть еще директории и так далее – мы должны достигнуть самого дна!

Пример списка со структурой директорий и файлов:

```
<ul>
  <li><i>file-1.txt</i></li>
  <li><i>file-2.txt</i></li>
  <li>
    <strong>directory-1</strong>
    <ul>
      <li><i>file-3.txt</i></li>
      <li><i>file-4.txt</i></li>
    </ul>
  </li>
</ul>
```

ЗАДАНИЕ 3 (ПОДСЧЕТ ПОСЕЩЕНИЙ)

1. Создать файл `visits.php`, который будет подсчитывать количество посещений сайта.
2. Открыть командную строку, перейти в ней в директорию, где находится файл и запустить команду `php -S localhost:8000`
3. Если при посещении ссылки <http://localhost:8000/visits.php> у пользователя не обнаружена cookie (`$_COOKIE`) с количеством посещений (например, с именем `visits`) - устанавливаем ему эту cookie (`setcookie`), а затем пишем - "Это ваш первый визит."
4. Если же при посещении <http://localhost:8000/visits.php> у пользователя найдена cookie (`$_COOKIE`) с количеством посещений - переустанавливаем эту cookie, увеличив на единицу старое значение (опять `setcookie`). Далее выводим строку - "Вы посещаете эту страницу в n раз.", где вместо n должно быть вставлено актуальное на тот момент количество посещений.